



Advanced Crafting for Invector

# Eadon Advanced Crafting for Invector TPC

Version 1.3

## Table of Contents

<b>Introduction .....</b>	<b>3</b>
Changelog.....	4
<b>Prerequisites .....</b>	<b>5</b>
<b>Configuration .....</b>	<b>6</b>
Recipe Category .....	6
Crafting Recipe .....	6
Location Type .....	8
Crafting Location.....	9
Crafting Recipe Manager .....	9
<b>Setting up a character to use Advanced Crafting .....</b>	<b>11</b>
Customizing the UI.....	13
<b>Configuring the Animator .....</b>	<b>15</b>
<b>Using Crafting Locations.....</b>	<b>17</b>
<b>Crafting Machines .....</b>	<b>18</b>

## Introduction

Eadon Advanced Crafting is an add-on for Invector's Third Person Controller asset for the Unity game engine. The goal of this asset is to provide advanced crafting functionality to any game implemented with Invector TPC.

The following functionalities are implemented:

- Crafting in game, not inside the inventory
- Based on Crafting Recipes
- Everything can be used to craft, no need for dedicated crafting ingredients
- Implicit recipes (known automatically by everyone)
- Explicit recipes which need to be found/bought
- Recipe categories for easy filtering
- Interruptible crafting
- If crafting is interrupted and the recipe does not allow interruptions, ingredients can be (optionally) lost
- Crafting by pressing a button or by holding a button
- Automatic success or random chance of success
- Crafting conditions
  - Optional requirement for tools
  - Optional requirement for location
- Integration with Eadon RPG for Invector
  - Optional requirement for race
  - Optional requirement for Class
  - Optional requirement for Alignment
  - Optional requirement for Skill
  - Optional requirement for Talent
- Custom animations while crafting, specific to the recipe
- Crafting machines

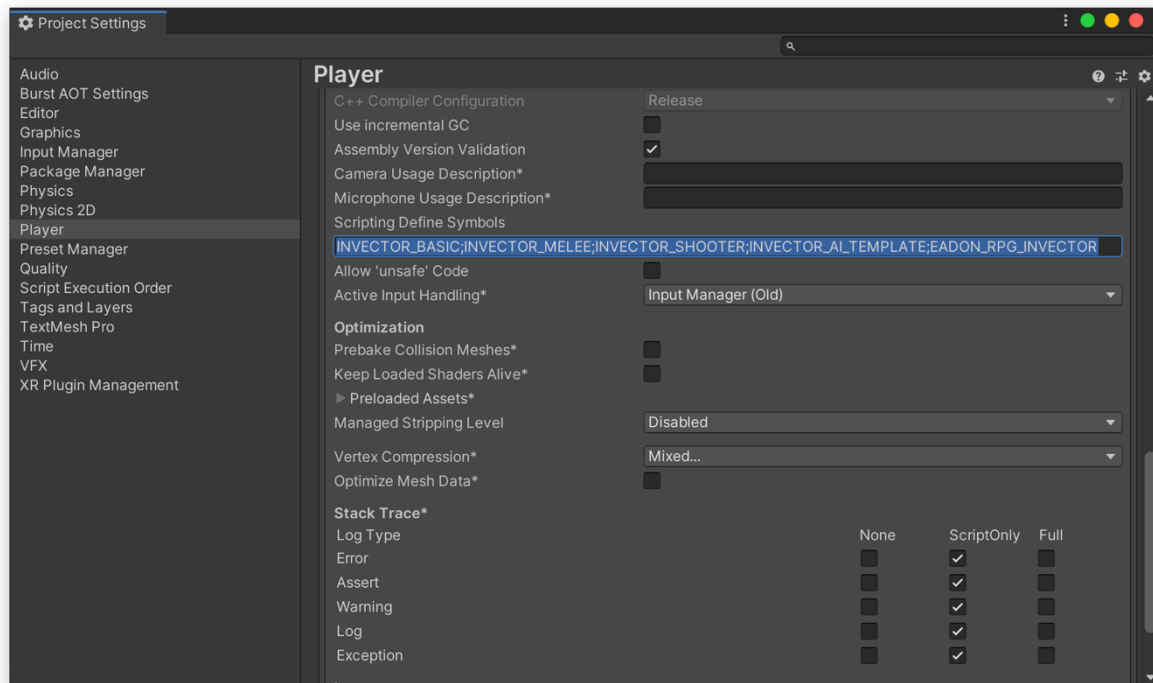
This add-on is currently compatible with Invector TPC version 2.5.3 and higher (Melee and Shooter) and tested on Unity 2018.4.30 and higher.

## Changelog

V 1.0	Initial release
V 1.1	Bug fixes and quality of life improvements
V 1.2	Added categories for recipes Configurable colors for UI messages
V 1.3	Added flag to determine whether to auto equip crafted items or not Added Crafting Machines (requires Eadon RPG for Invector)

## Prerequisites

Eadon Advanced Crafting has support for Eadon RPG for Invector, which is automatically enabled if Eadon RPG for Invector v1.4 is installed. To enable support for previous versions, please add `EADON_RPG_INVECTOR` to the scripting defines list in Player options:



In addition, several errors will appear as soon as the add-on is imported because the code relies on the presence of a `ItemType`. In order to fix this error, you need to go to `Invector/Inventory/Item Enums/Open ItemEnums Editor` and click on “REFRESH ITEMENUMS” at the bottom.

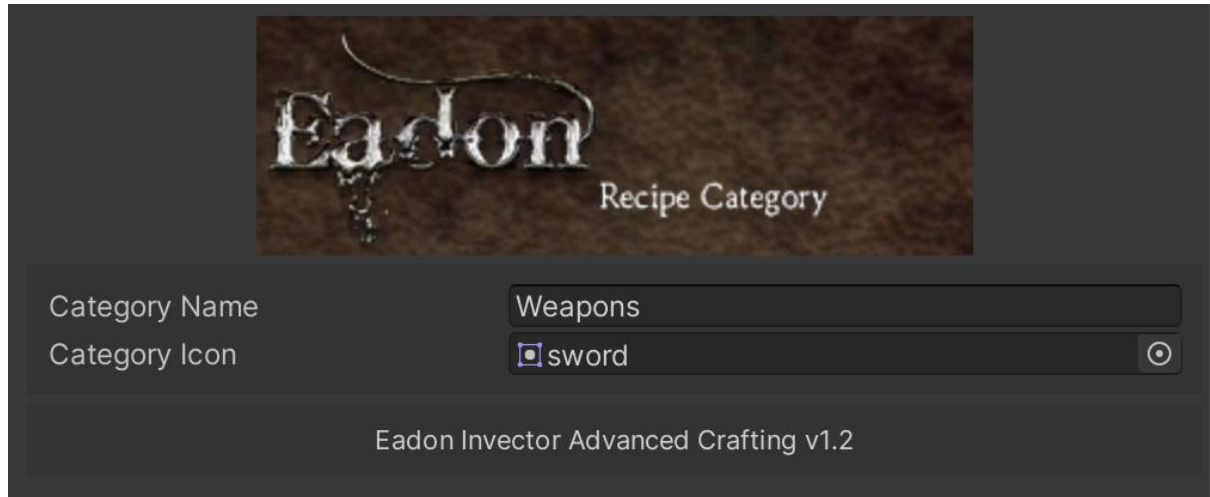
Eadon Advanced Crafting requires TextMesh Pro, make sure it’s installed (with the Essential Resources as well). In recent versions of Unity, it comes preinstalled (but the essential resources need to be installed from the Window -> TextMesh Pro menu), in older versions it needs to be added manually.

## Configuration

Before the crafting system can be used, it must be configured. Eadon Advanced Crafting is configured through a set of ScriptableObjects that determine what is available to the system. All Eadon Advanced Crafting ScriptableObjects can be created using commands found under the **Assets -> Create -> Eadon Crafting** menu.

### Recipe Category


Recipe Categories allow for easy filtering of recipes in the UI. They look like this:



The two fields are the name of the category and the category icon, to be displayed in the UI (see below).

### Crafting Recipe

Crafting Recipes are the core of the system as they define what can be crafted and how. They look like this:



**Recipe**

Recipe Name

Recipe Icon

Recipe ID

Category

Is Implicit
☒

Can Be Interrupted
☐

Consume If Not Finished
☐

Crafting Time

**Ingredients**

▼ Ingredients

Size

▼ Element 0

V Item Id

Quantity

▼ Element 1

V Item Id

Quantity

**Result**

Automatic Success
☐

Success Chance

Output Item ID

Output Amount
**Conditions**

Requires Tools
☐

Requires Locations
☐
**Animations**

Start Animation
**Particles**

Craft Particles

Particle Offset



Eadon Invector Advanced Crafting v1.2

The fields are as follow:

Field name	Purpose
<b>Recipe Name</b>	The name of the recipe, used by the UI
<b>Recipe Icon</b>	The icon of the recipe, used by the UI
<b>Recipe ID</b>	The ID of the recipe, link to the vItem
<b>Category</b>	The category of the recipe

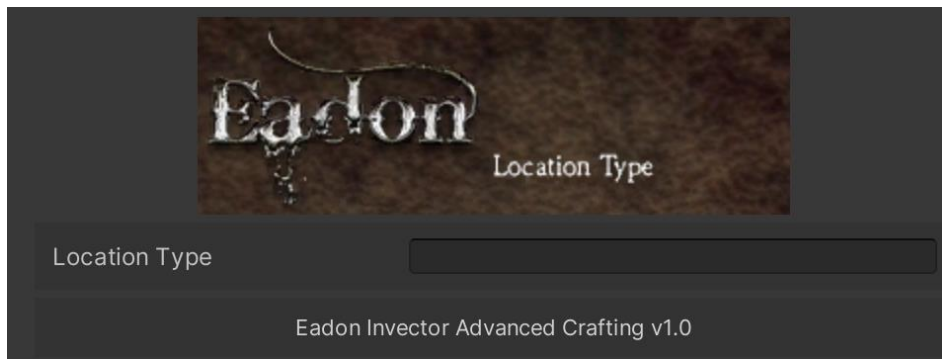
<b>Is Implicit</b>	An implicit recipe is known by everyone and doesn't need to be in the player inventory
<b>Can Be Interrupted</b>	A flag to indicate if the recipe allows interrupting and resuming
<b>Consume If Not Finished</b>	A flag to indicate if the ingredients are consumed if the crafting is interrupted
<b>Crafting Time</b>	How long it takes to craft, in seconds
<b>Ingredients</b>	An array of vitem id and quantity for every component required
<b>Automatic Success</b>	A flag to indicate if crafting is automatically successful or not
<b>Success Chance</b>	The chance of successful crafting, only appears if automatic success is disabled
<b>Output Item ID</b>	The vitem ID of the resulting item
<b>Output Amount</b>	The amount of output item crafted
<b>Requires Tools</b>	A flag to indicate if tools are required to craft. Tools are not consumed
<b>Tools ID</b>	A list of vitem IDs for the tools. Only visible if <b>Requires Tools</b> is selected
<b>Requires Locations</b>	A flag to indicate if a specific location is required for this recipe to be crafted
<b>Location Types</b>	A list of LocationTypes (see below). Only visible if <b>Requires Locations</b> is selected
<b>Requires Race *</b>	A flag to indicate if a specific race is required for this recipe to be crafted
<b>Required Race *</b>	The required race. Only visible if <b>Requires Race</b> is selected
<b>Requires Class *</b>	A flag to indicate if a specific class is required for this recipe to be crafted
<b>Required Class *</b>	The required class. Only visible if <b>Requires Class</b> is selected
<b>Requires Alignment *</b>	A flag to indicate if a specific alignment is required for this recipe to be crafted
<b>Required Alignment *</b>	The required alignment. Only visible if <b>Requires Alignment</b> is selected
<b>Requires Skill *</b>	A flag to indicate if a specific skill is required for this recipe to be crafted
<b>Required Skill *</b>	The required skill. Only visible if <b>Requires Skill</b> is selected
<b>Requires Talent *</b>	A flag to indicate if a specific talent is required for this recipe to be crafted
<b>Required Talent *</b>	The required talent. Only visible if <b>Requires Talent</b> is selected
<b>Start Animation</b>	The animation to play when crafting starts

All the fields marked with \* will appear only if Eadon RPG for Invector is installed in your project.

### Location Type

Location Types are like tags for locations. They are used to mark the types of locations so that multiple crafting locations can share the same type. They look like this:

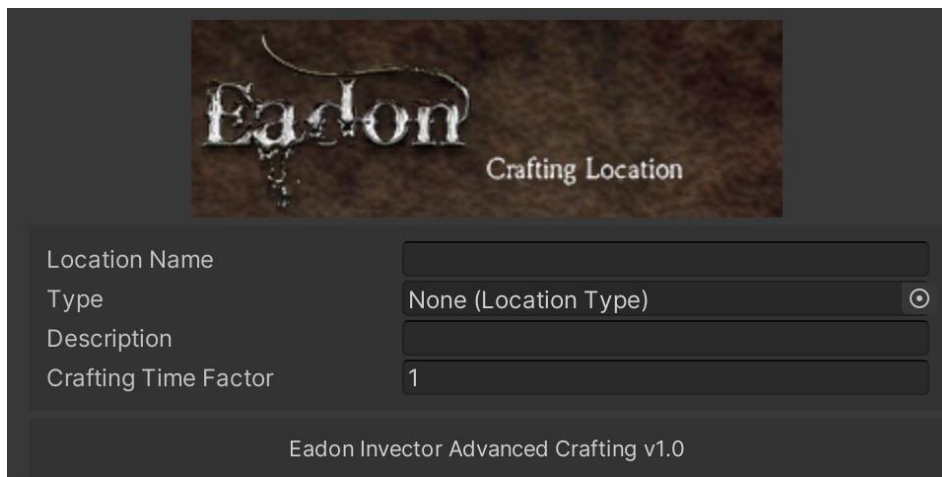




The field is a freeform string.

### Crafting Location

Crafting Locations represent places where specific crafting activities can take place, such as forges, alchemist labs, workbenches etc. They look like this:

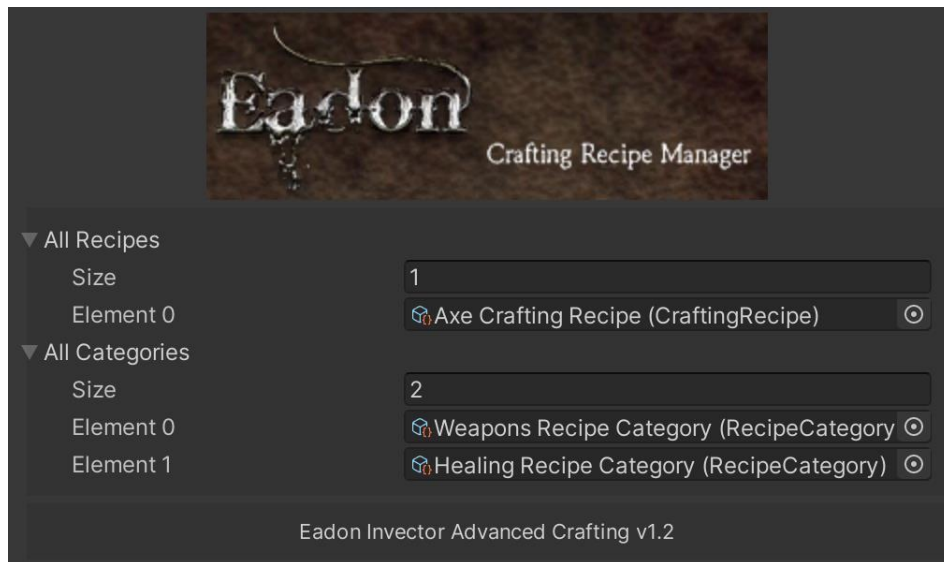


The fields are:

Field name	Purpose
<b>Location Name</b>	The name of the location
<b>Type</b>	The LocationType of this location
<b>Description</b>	A description of this location
<b>Crafting Time Factor</b>	A multiplier for the crafting time (less than 1 speeds up, more than 1 slows down), useful to determine “quality” of the location

### Crafting Recipe Manager

The Crafting Recipe Manager is a scriptable object which stores a reference to all Crafting Recipes and all Recipe Categories available. It looks like this:

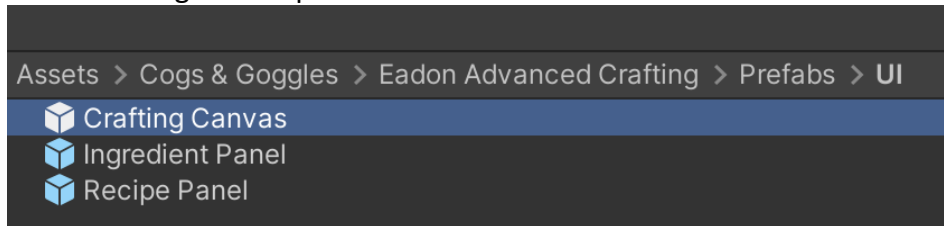


When you create a new recipe or category, it needs to be added here in order for it to be available in game.

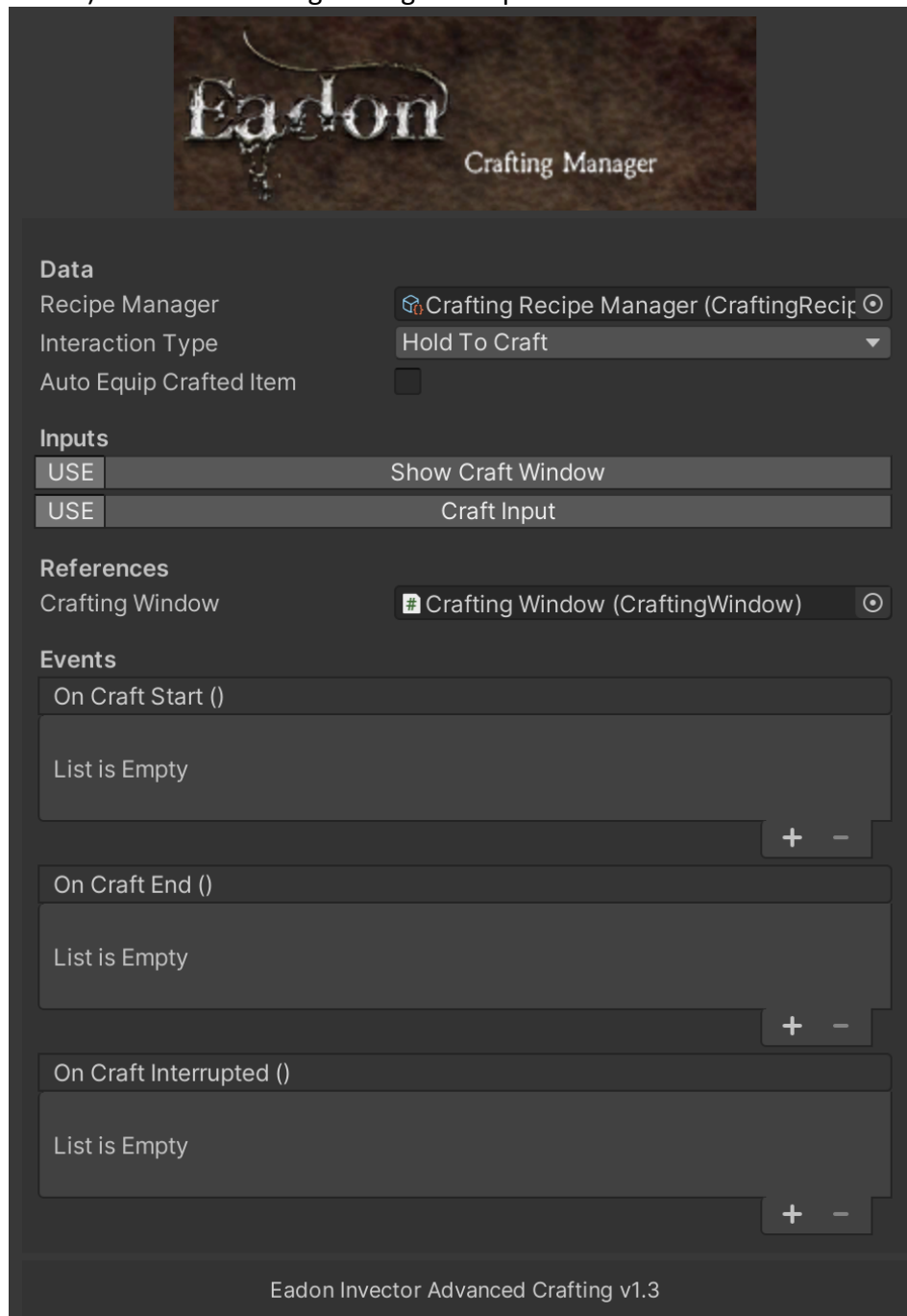
## Setting up a character to use Advanced Crafting

Setting up a character to use this addon is very simple and requires two steps:

- 1) Add the Crafting Canvas prefab to the character:



- 2) Add the Crafting Manager component to the character:



The fields are:

Field name	Purpose
<b>Recipe Manager</b>	The Crafting Recipe Manager to use
<b>Interaction Type</b>	Choice of Simple Press and Hold to Craft
<b>Auto Equip Crafted Item</b>	A flag to auto equip crafted items or not
<b>Show Craft Window</b>	The input to show the craft window, by default mapped to the L key
<b>Craft input</b>	The input to actually craft, by default mapped to the K key
<b>Crafting Window</b>	A reference to the crafting window object (child of the crafting UI prefab)
<b>On Craft Start</b>	An event triggered when crafting starts
<b>On Craft End</b>	An event triggered when crafting ends
<b>On Craft Interrupted</b>	An event triggered when crafting is interrupted

The crafting UI (which is meant to be customised based on your UI standards) looks like this:



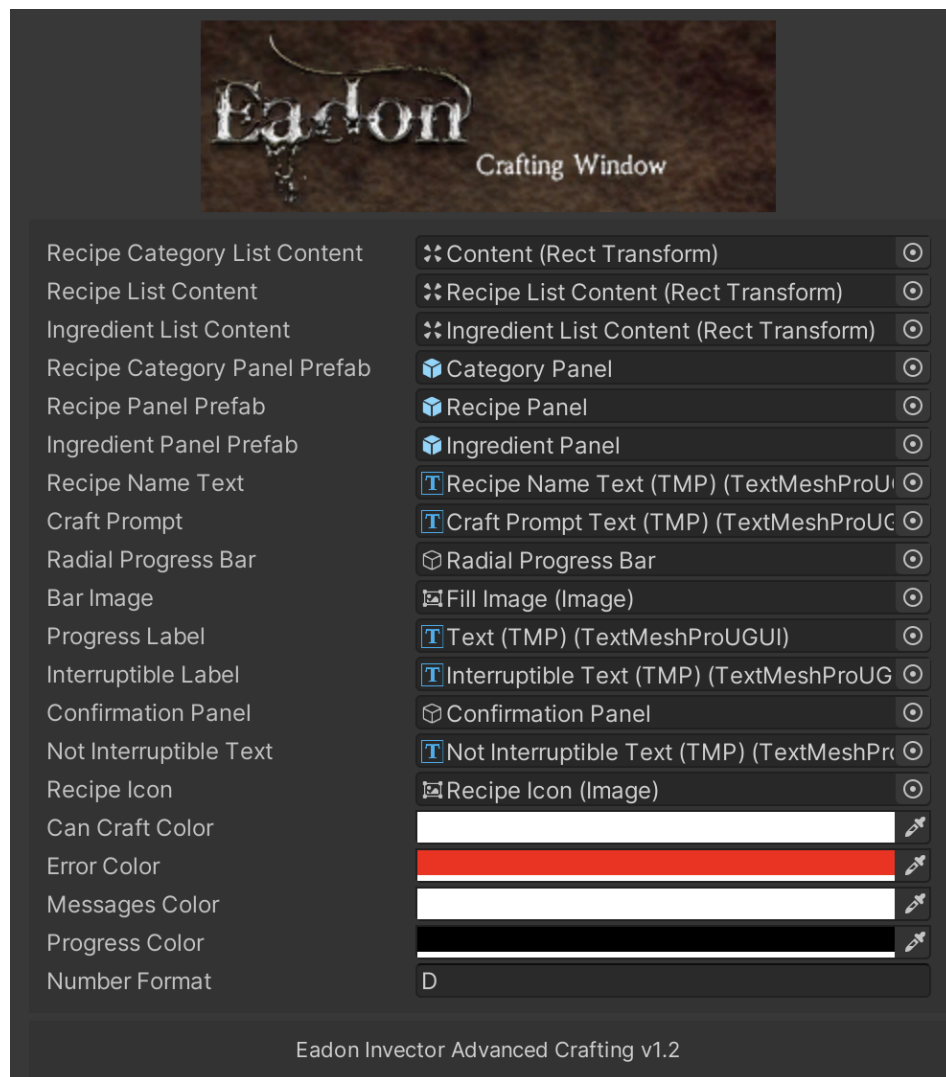
On the left is a list of all the categories of recipe defined in the system and the list of recipes the player has access to. This list includes all implicit recipes and all the explicit recipes in his inventory, filtered by the selected category. When active, the mouse cursor appears, and clicking on a recipe will tell you which ingredients are needed.

If all the ingredients are available, a prompt appears to let you start crafting.

The Crafting Manager listens to the `vThirdPersonController` and interrupts the crafting if the character is hit.

## Customizing the UI

The UI is handled through a master component, called CraftingWindow. It's located in the first child of the Crafting Canvas prefab and looks like this:



The customizable fields are the 5 bottom ones, the others are references to components or instantiable prefabs (see below).

The fields are:

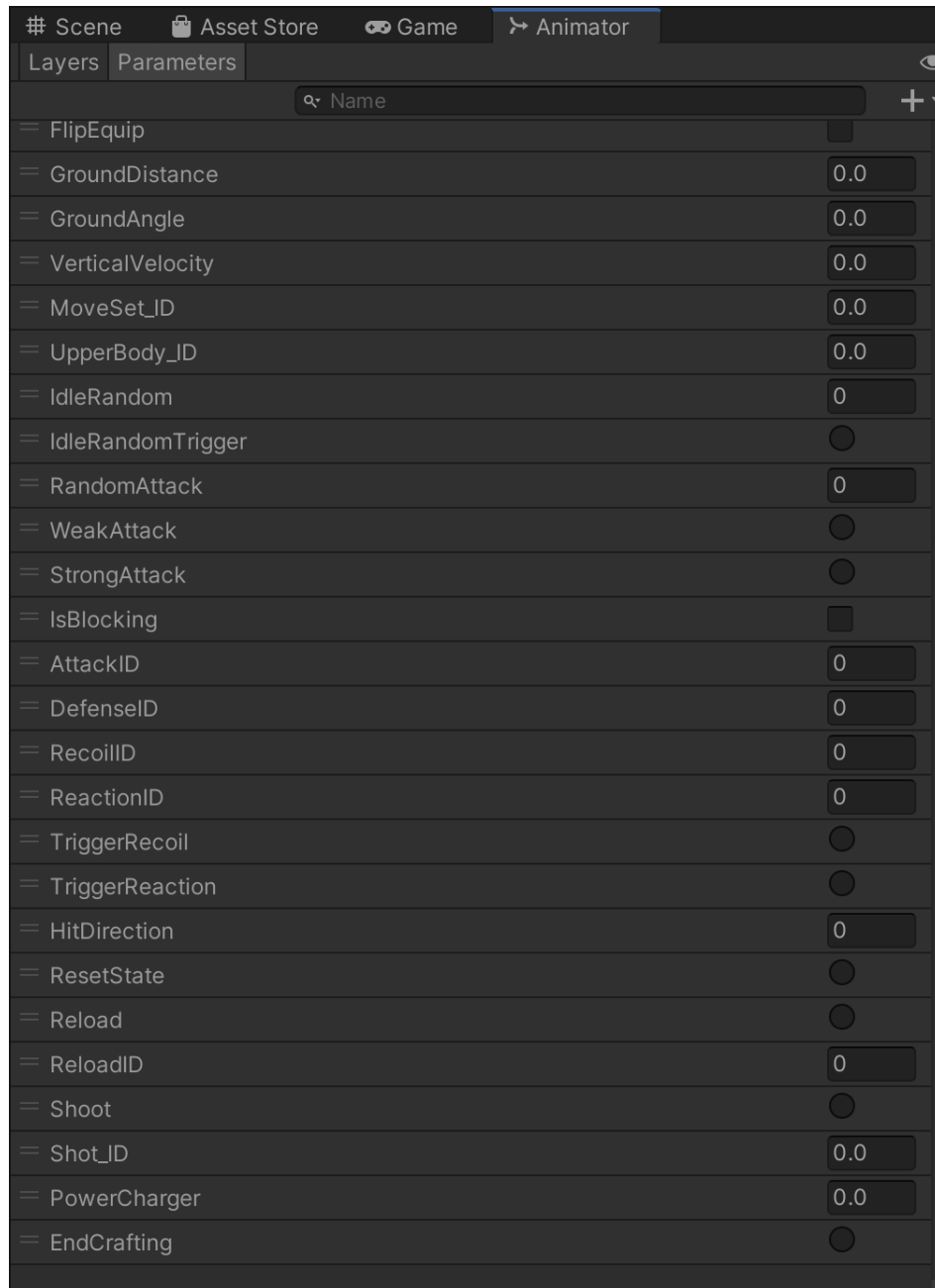
Field name	Purpose
<b>Can Craft Color</b>	The color of the message indicating that a recipe can be crafted
<b>Error Color</b>	The color of the various error messages
<b>Messages Color</b>	The color of all the other messages
<b>Progress Color</b>	The color of the crafting progress indicator
<b>Number Format</b>	The format of the progress percentage. Use D for an integer (no decimals) of Fx for decimals (where x is the number of decimal digits to show)

The various lists display categories, recipes and ingredients by instantiating panel prefabs of the appropriate type. These prefabs are located in Assets/Cogs & Goggles/Eadon Advanced Crafting/Prefabs/UI and all contain a mandatory script holding references to the various needed fields.

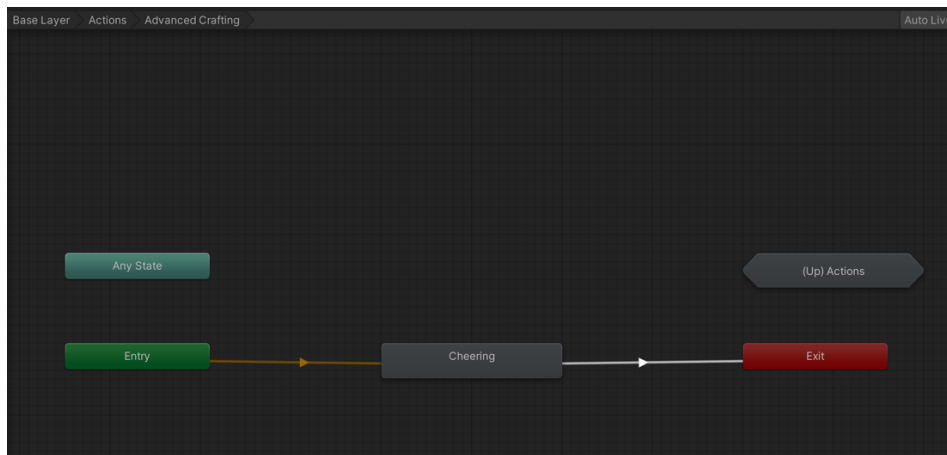
These prefabs can be customized freely AS LONG AS the component stays in place and holds valid references. You can add more images, labels, subpanels, etc

## Configuring the Animator

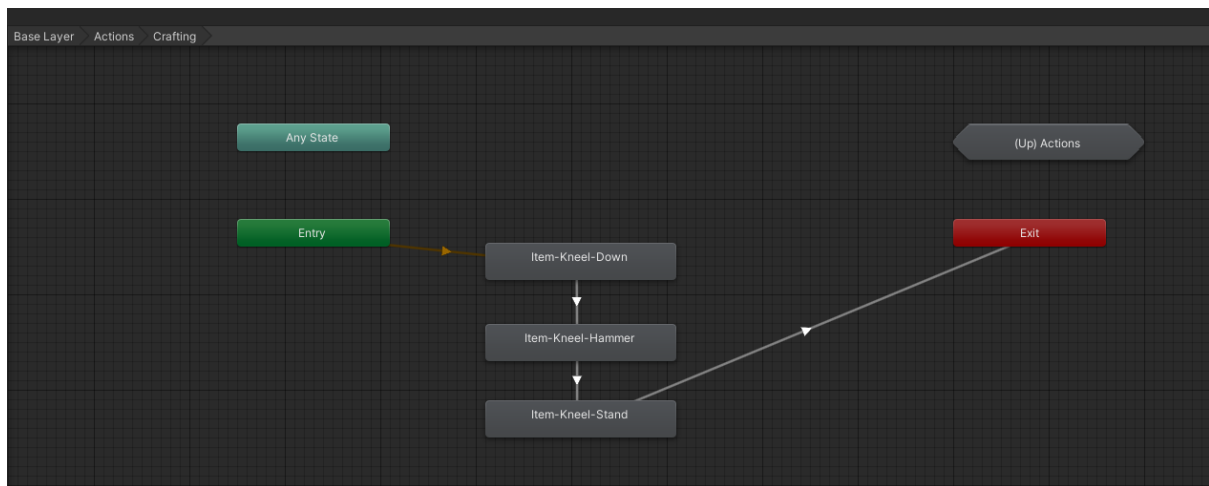
A Crafting Recipe lets you specify a custom animation to be played while the character is crafting. If the field is not empty, as soon as the character starts crafting, the corresponding animation is played. In order for this system to work, you need to add a trigger named EndCrafting to your animator controller:



The example animator comes with a simple “Cheering” animation set up like this:



But this could be expanded further, for example in a setup similar to this:

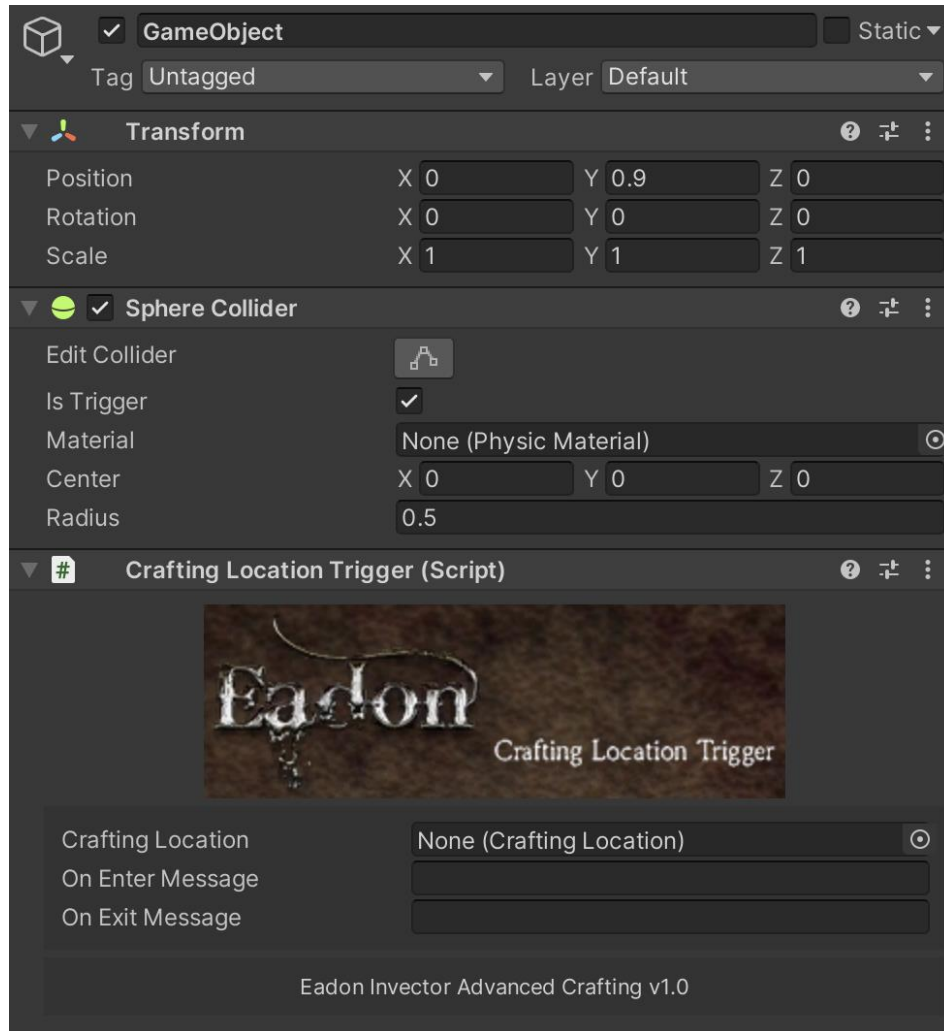


Where the start animation points to a kneeling animation which transitions to a looping hammering animation and the trigger transitions to a stand-up animation which goes then back to locomotion. Please note that this example is not included in the asset.



## Using Crafting Locations

Crafting Location are easy to use. You just need to pick the scene element (or create an empty game object) and add a suitably sized trigger collider and the Crafting Location Trigger. It looks like this:



The fields are:

Field name	Purpose
<b>Crafting Location</b>	The Crafting Location scriptable object for this location
<b>On Enter Message</b>	A message displayed via the vHUDManager at the top of the screen when the character enters the crafting location
<b>On Exit Message</b>	A message displayed via the vHUDManager at the top of the screen when the character exits the crafting location

The trigger will set the location on the Crafting Manager automatically.

## Crafting Machines

Crafting Machines are a new feature which require the presence of **Eadon RPG for Invector** in your project as it leverages the stash system introduced in that addon.

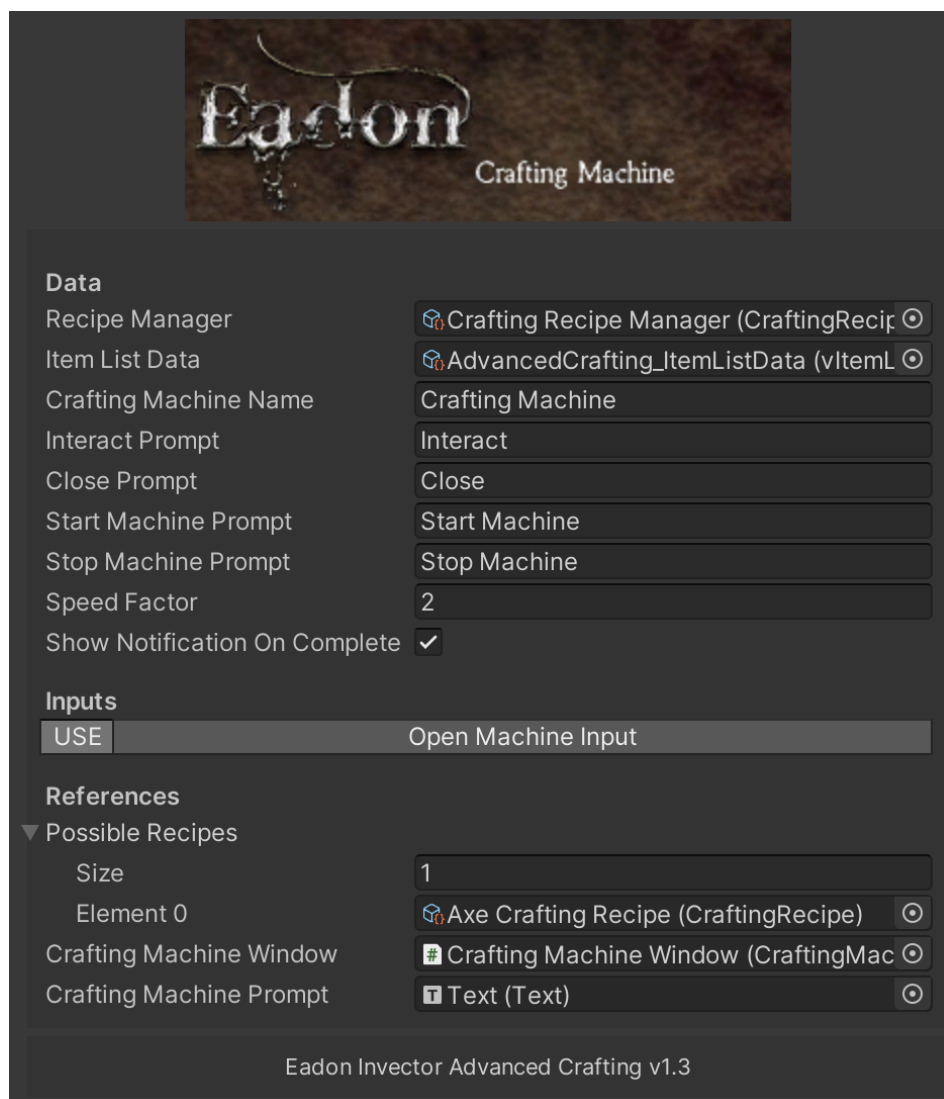
A crafting machine is a combination of a stash (used to dump ingredients into the machine and retrieve crafted items) and a standalone crafting engine.

In order to use a crafting machine, you need to setup the stash system as per the documentation of **Eadon RPG for Invector**.

After that, create a game object and do the following:

- 1) Add a stash as a child object
- 2) Add a Crafting Machine Canvas prefab as a child
- 3) Add a Crafting Machine Trigger prefab as a child
- 4) Add a EadonCraftingMachine component

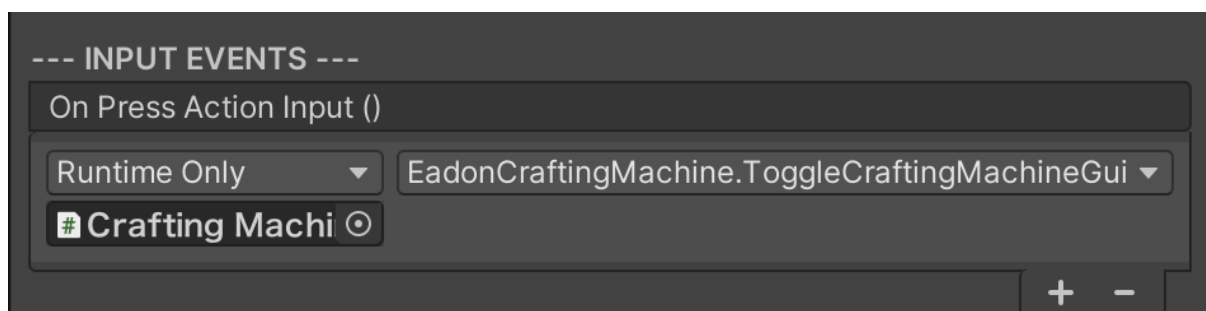
The EadonCraftingMachine component looks like this:

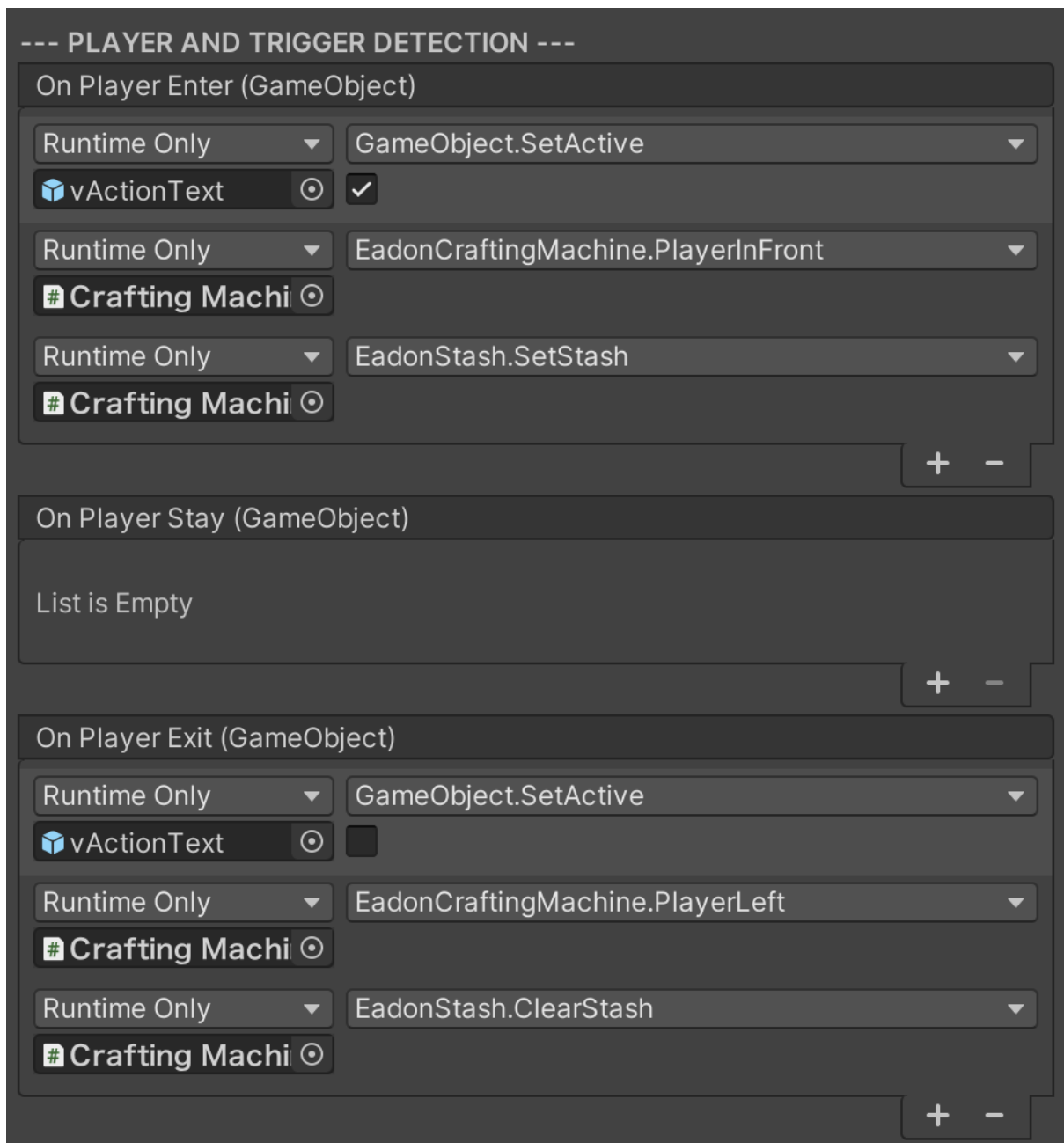


The fields are:

Field name	Purpose
<b>Crafting Location</b>	The Crafting Location scriptable object for this location
<b>Recipe Manager</b>	A reference to your recipe manager
<b>Item List Data</b>	A reference to your vItemListData
<b>Crafting Machine Name</b>	The name of this crafting machine (used in the UI and the notifications)
<b>Interact Prompt</b>	The interaction prompt message to open the UI
<b>Close Prompt</b>	The interaction prompt message to close the UI
<b>Start Machine Prompt</b>	The interaction prompt message to start the machine
<b>Stop Machine Prompt</b>	The interaction prompt message to stop the machine
<b>Speed Factor</b>	A multiplier for the standard crafting time defined in the recipe. Values greater than 1 increase the crafting time, while values lower than 1 reduce the time
<b>Show Notification On Complete</b>	A flag to indicate if the machine should notify finishing crafting something in the vHUDController text display area
<b>Open Machine Input</b>	The GenericInput to use to open the machine stash interface
<b>Possible Recipes</b>	A list of all the CraftingRecipes the machine can craft
<b>Crafting Machine Window</b>	A reference to the CraftingMachineWindow component (child of the Crafting Machine Canvas prefab)
<b>Crafting Machine Prompt</b>	A reference to the Text component of the vActionText child of the Crafting Machine Trigger

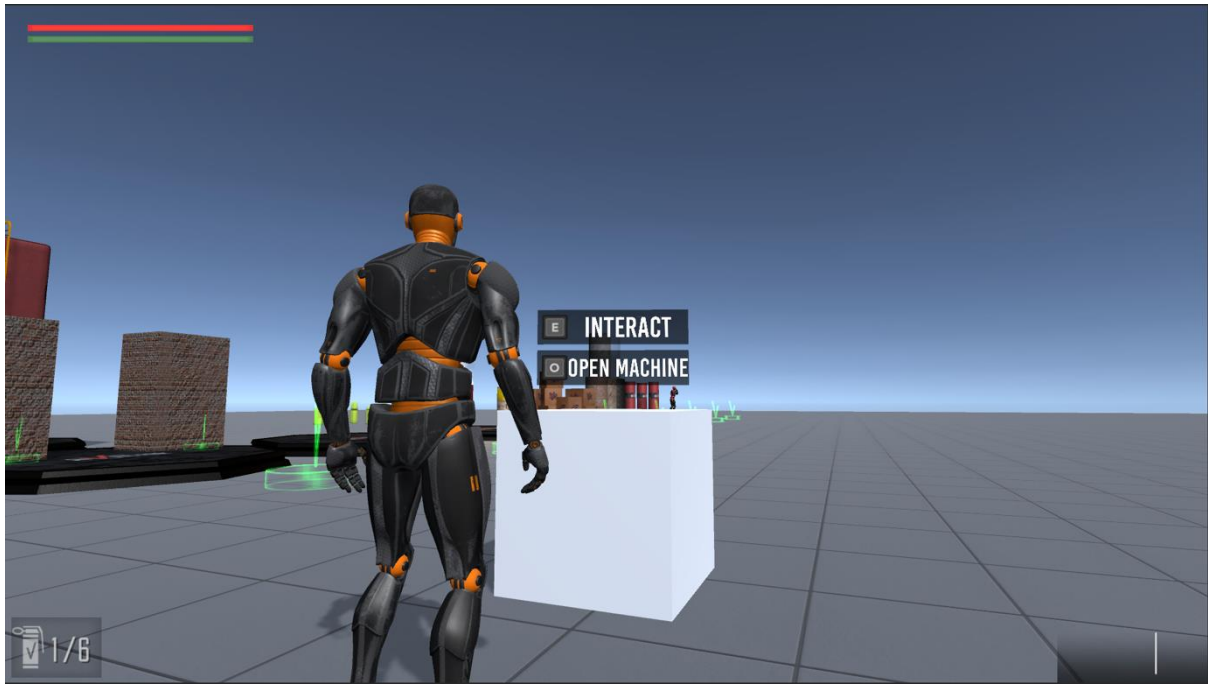
The Crafting Machine Trigger prefab contains a vTriggerGenericAction component which needs the following events configured:





You can see an example in the Crafting Machine Demo scene.

When the player approaches the crafting machine, two prompts appear:



The bottom one opens the stash interface to transfer ingredients and finished products between the player and the machine. The top one opens the machine interface, which looks similar to and works the same way as the in game crafting interface:



If the machine has enough ingredients, you can place items in the queue. Queueing items removes the ingredients from the stash. Clearing the queue puts the ingredients back in the stash.

Once elements are in the queue, the machine can be started. If you open the interface when the machine is running, you can stop it. If you stop a machine, the recipe you're currently

crafting is interrupted and the ingredients are either placed back in the stash or destroyed depending on the settings on the recipe.